

ISD-VPE15100

User's Manual

ISD15100 Series

Multi-Message Record/Playback Devices

with Digital Audio Interface

1.	Quick Start	3
1.1	Standalone Operations.....	3
1.2	Install the VPE15100	4
1.3	Browse the Example Project	5
2.	Build the Example Project	7
2.1	Create a New Project.....	7
2.2	Device/Clocks	8
2.2.1	Advanced Clock Configuration	9
2.3	Voice Prompt.....	10
2.3.1	Keystrokes to Re-order the VPs.....	11
2.4	Reserved Memory	12
2.5	Voice Macro	13
2.6	Configuration	16
2.7	Memory Map.....	17
2.8	Archive and Clean.....	18
3.	Evaluation Board Overview.....	20
3.1	Connect a uC to the Evaluation Board	22
3.2	Program the ISD15100.....	23
3.3	The Programmer Daughter Card – ISD-ES15100_PROG	24
3.4	The Demo Board – ISD-DEMO15100.....	24
4.	USB Emulation	26
4.1	Audio Cmds.....	26
4.2	Digital Cmds	28
4.3	Enriched Commands & USB Logs	30
5.	Revision History	31

1. Quick Start

Different from the analog ISD ChipCorders®, the ISD15100 family is digital. As the ISD15100 is a project-based chip, users need the VPE15100 software utility to build an ISD15100 project and generate a binary programming file, which will be digitally programmed into the ISD15100 chip.

The VPE15100 software utility provides an easy evaluation of the ISD15100 family. It serves three main purposes: (1) Create scenarios in a user-defined memory structure; (2) Ease the hassle of configuring the ISD15100 signal paths, volume controls, AGC, I²S ... etc; (3) Provide a complete evaluation of compression algorithms. The ISD15100 offers a variety of compression algorithms at various bit rates and sampling rates to fit various applications' needs. The algorithms include Enhanced ADPCM, VBR (variable-bit-rate) ADPCM, uLaw, DuLaw (differential uLaw), and PCM.

The Voice Prompt editing panel allows users to evaluate the audio qualities with different compression algorithms and sampling rates; likewise, the Voice Macro editing panel enables users to edit the audio scripts like composing a sentence by concatenating the voice prompts. The path-view on the Configuration panel makes it easy to configure the signal paths for Record from various inputs or Playback to various outputs. The 'Reserved Message/User Data' editing panel allows user to reserve memory area for future message recordings or for digital data storage. After the project is completed, the VPE15100 generates a straight binary programming file (the .mem file in the project folder), that can be programmed into ISD15100 memory.

The VPE15100 is a GUI-based program run in Microsoft Windows® environments.

Getting Started

The evaluation kit contains the following:

- A Getting Started sheet that guides users to our website (<http://www.nuvoton-usa.com/ISD15100/vpe>) to download the latest VPE15100 software utility.
 - o Users will need to fill out a short form on the webpage to get the username and password required for the installation of VPE15100.
- A USB evaluation board (ISD-ES15100_USB).
- A mini-USB cable.
- An 8-ohm speaker.

Equipment requirements:

⇒ PC running Windows Vista, XP, NT, or 2000.

1.1 Standalone Operations

The USB evaluation board can either be operated standalone or controlled by the VPE15100. To operate standalone:

1. Make sure the VPE15100 project has been properly programmed into the board.
 - a. By default, the USB evaluation board comes with an example project that sets everything correctly for the standalone operations.
2. Make sure the playback path is set in the PU macro (VM1).
3. Attach the 8-ohm speaker to the audio jack J16.
4. Attach the keypad to the pin-header HJ1.

5. Insert 3 AAA batteries into the battery holder underneath the board.

In 3 seconds, the USB controller will power on the board and reset the on-board ISD15108 chip. Then, press the keypad:

- ExeVM: power-up the on-board ISD15108 chip or send the ExeVM command.
 - The USB controller automatically checks if the on-board ISD15108 chip is powered up. If not, it sends the power-up command; if yes, it sends the ExeVM command.
- STOP: send STOP command and clear the index.
- 0 ~ 9: indexes for executing VM.
- VOLUP/VOLDN: increase/decrease the volume.

For example, to execute VM 25, do the following:

- Press ExeVM. ISD15108 will be powered-up.
- Press 2. Users shall see '2' displayed on the 7-segment display.
- Press 5. Users shall see '5' displayed on the 7-segment display.
- Press ExeVM to execute VM 25.

1.2 Install the VPE15100

Be sure to install the VPE15100 software utility before connecting the USB evaluation board to PC.

After the VPE15100 is properly installed, plug in the evaluation board to PC; then, users shall see a message popped up as Figure 1-1. Select 'install from a list or specific location (Advanced)', and then click 'Next'.



Figure 1-1 Install USB Driver – 1

The USB driver is located in the VPE15100 installation folder, which by default is "c:\Program Files\ISD-VPE15100\EmDriver\" as shown in Figure 1-2. Click 'OK'.



Figure 1-2 Install USB Driver – 2

Users shall see the installing of USB driver in progress as Figure 1-3.

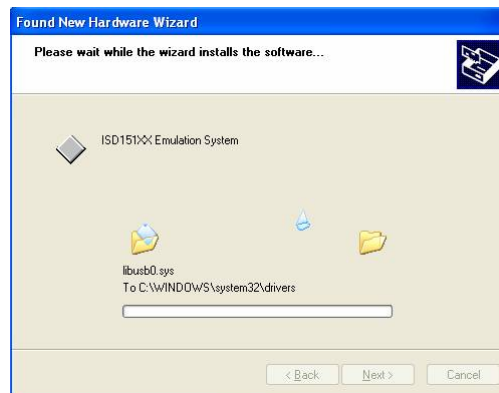


Figure 1-3 Install USB Driver – 3

After the installation is complete, reset the USB evaluation board either by unplugging and plugging in the board or by pressing the RESET_USB button.

1.3 Browse the Example Project

By default, the USB evaluation board comes with an example project. While users install the VPE15100, the example project is automatically copied onto PC at the VPE15100 installation folder, which by default is "c:\Program Files\ISD-VPE15100". The project name is ISD15100example.

Users can browse the example project to quickly get a sense of the VPE15100 project structure:

- Open the project:
 - o Project -> Open; load the ISD15100example.prj.
- Browse the panels on VPE15100 to gather basic information about the example project:
 - o The clock source is set to 4.096 MHz crystal. Users may have noticed that there is a 4.096 MHz crystal on the board.
 - o There are several voice prompts.
 - Users can try different algorithms and sampling rates to experience the sound quality.

- o Several empty messages are reserved.
- o Several voice macros are composed.
 - The POI macro (voice macro 0) is composed as:
 - Set clock source to 4.096 MHz crystal (0x27).
 - Set a playback path to PWM.
 - Play one beep sound.
 - Power down.
 - The PU macro (voice macro 1) is composed as:
 - Set clock source to 4.096 MHz crystal (0x27).
 - Set a playback path to PWM.
 - Play two beep sounds.
 - Idle.
- Go to the Memory Map screen, and users shall see the memory map of the example project. If not, users can click the “Create Programming File” button to generate the memory map as well as the binary programming file.
 - o The binary programming file is the .mem file in the project folder.
 - o As long as the USB evaluation board is connected to PC, the button “Burn Device” is effective. Click it so that the VPE15100 will:
 - Chip-erase the on-board ISD15108 chip.
 - Program (digital-write) the binary programming file into the chip.
 - Send a high pulse to the RESET pin to reset the on-board ISD15108 chip.
- Go to the ‘Audio Cmds’ screen and try those voice prompts and voice macros.

2. Build the Example Project

This section introduces the procedure of making the example project.

2.1 Create a New Project

Launch the VPE15100, and users shall see their username (A) displayed on the subject bar. Please note that the username is part of the VPE15100 project, which will be encrypted into the binary programming file when the VPE15100 generates it.

On the menu bar, click 'Project' (B) -> 'New', and users shall see a pop-up window as Figure 2-2. Type in the desired project name (C) and assign a project location (D). Click 'OK' (E), and users shall see the project path shown on the subject bar (F). In the assigned location users shall see that a project folder has been created.

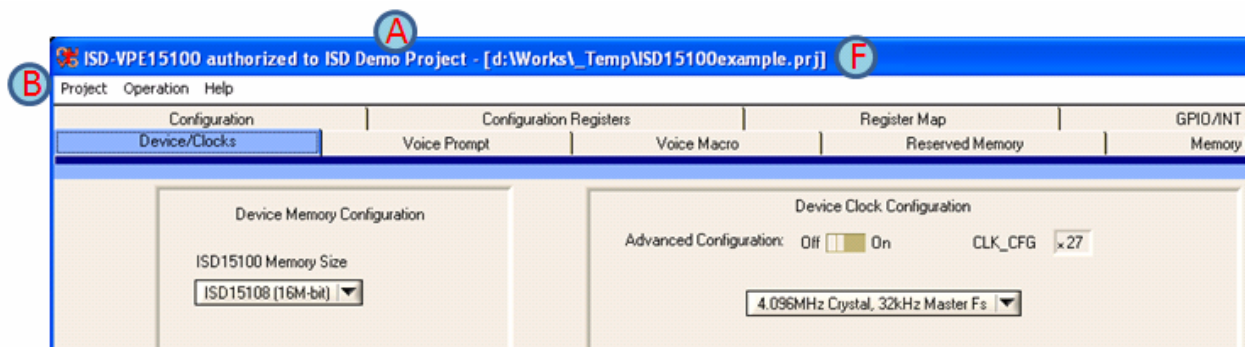


Figure 2-1 Username & Project Path shown on the Subject Bar

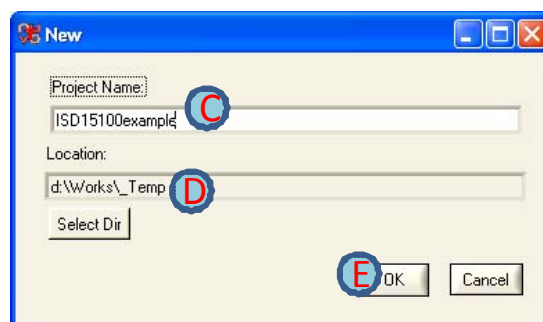


Figure 2-2 Create a New Project

There are three subfolders in the project folder:

- BinFiles: where the compressed VPs are stored.
- WavFiles: where the re-sampled and de-compressed VPs are stored.
- WavSrc: where wave sources of VPs are stored.

2.2 Device/Clocks

Users specify the device (A) and the clock source (B) on the 'Device/Clocks' screen. The Memory Size selection (A) is for the VPE15100 to calculate and show the memory usage (C). If the USB evaluation board is connected to PC, the VPE15100 automatically sets the device (A) to ISD15108.

The VPE15100 provides several common clock settings. For every clock setting, seven sub-sampling rates are available for VP compression, as shown in Table 2-1. Please note that the clock setting is part of the VP compression, so whenever users change the clock setting, they have to 'Re-compress All' VPs (a button on 'Voice Prompt' screen). The one-byte CLK_CFG (D) is for users to set the clock in the POI/PU voice macros. If the common clock settings that the VPE15100 provides do not fit the users' needs, users may switch to the advanced view (E).

Please note that the clock setting (B) here is ONLY for the VP compressions. Users also need to set the clock for the ISD15100 chip by either setting it in their PU macro or by sending the SPI command SET_CLK_CFG after the ISD15100 chip is powered up.

The default clock setting for the ISD15100 chip when POI/PU:

- If the POI/PU macro is empty:
 - Clock sets to 0x34, which is internal oscillator with internal resistor.
- If the POI/PU macro exists, but users do not set the clock source in it:
 - Clock sets to 0x00, which bypasses the PLL.

In the example project, we choose ISD15102 (A) and set the clock source to 4.096 MHz crystal (B).

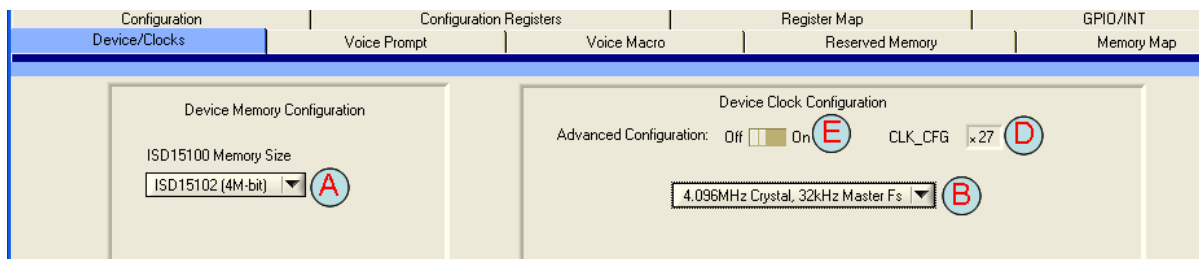


Figure 2-3 Device/Clocks Setting



Figure 2-4 Memory Usage Indicator

Sub-Sampled ratio	Master Sample Rate Fs (kHz)			Available Sample Rates (kHz)
	32	44.1	48	
8	4	5.5125	6	
6	5.333	7.35	8	
5	6.4	8.82	9.6	
4	8	11.025	12	
2.5	12.8	17.64	19.2	
2	16	22.05	24	
1	32	44.1	48	

Table 2-1 Sub-sampling Rates

2.2.1 Advanced Clock Configuration

The Clock Source (A) includes four clock sources:

- Internal oscillator,
- Internal oscillator with external resistor,
- I²S clock, and
- Crystal.

The selected clock will feed into the phase-lock-loop (PLL) (B) for generating the internal master clock (MCLK). Please refer to Table 2-2 for PLL M & N settings for a required master sampling rate. For every clock setting, there are seven available sub-sampling rates for VP compression (C).

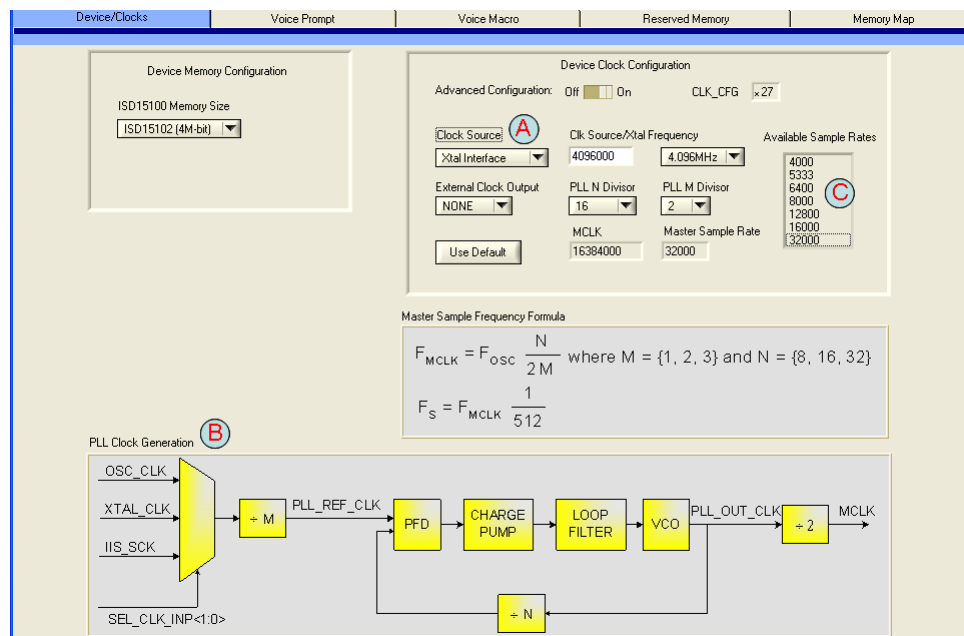


Figure 2-5 Advanced Clock Configuration

Xtal (MHz)	M	N	Fs (kHz)
1.024	1	32	32
2.048	2	32	32
3.072	3	32	32
4.096	2	16	32
6.144	3	16	32
8.192	2	8	32
12.288	3	8	32
1.4112	1	32	44.1
2.8224	2	32	44.1
4.2336	3	32	44.1
5.6448	2	16	44.1
8.4672	3	16	44.1
11.2896	2	8	44.1
16.9344	3	8	44.1
1.536	1	32	48
3.072	2	32	48
4.608	3	32	48
6.144	2	16	48
9.216	3	16	48
12.288	2	8	48
18.432	3	8	48

Table 2-2 PLL M & N Settings

If users need really high quality, they might want to try 44.1 kHz sampling rate. According to Table 2-2, users can use an 11.2896 MHz crystal and set the PLL as M = 2, N = 8. In this case users also have to change the crystal on the USB evaluation board.

If the input Clock Source is set to internal Oscillator, the clock rate is fixed at 2.048 MHz. If the input clock source is set to Internal Oscillator with External Resistor, the clock rate varies with the resistor value. User can type in different resistor values to see the calculated clock rate. Users can change the N and M of the PLL to generate the desired master clock (MCLK) and thus the available sampling rates.

2.3 Voice Prompt

Voice prompts are pre-recorded audio of any length, from short words, phrases or sound effects to long passages of music. These Voice Prompts can be played back in any order as determined by the users and applications. A voice prompt consists of two components:

- (1) An index pointing to the pre-recorded audio.
- (2) The pre-recorded audio.

Before adding voice prompts, first set the default sampling rate (A) and the compression algorithm (B).

To add voice prompts, follow the steps below:

- (1) Set sampling rate (C) and compression algorithm (D) for the VPs you are going to add. Users can change the individual VP's sampling rate (E) and/or compression algorithm (F) later when evaluating the quality.
- (2) Add voice prompts (G).
- (3) Repeat steps (1) & (2).
- (4) Change the VP tags (H) if needed; by default, the tags are the filenames.
- (5) Re-order the VPs if needed; select VP by clicking the square (I), and click the up/down arrows (J) to re-order the VPs.

If users change the clock configuration on 'Device/Clocks' screen, they must re-compress all VPs (K) because the clock setting is part of the header of a VP.

Users can evaluate the VPs through PC (software simulation) or through the USB evaluation board (the real hardware):

- Through PC (L):
 - i. Original voice prompts before re-sampling & compression.
 - ii. Re-sampled but not yet compressed voice prompts.
 - iii. Compressed voice prompts.
- Through the USB evaluation board:
 - i. Plug in the USB evaluation board and the 'Play to device' button (M) shows up. Click the 'Play to device' button (M).
 1. Before playing the VPs, users need to set a playback path. Users can switch to the 'Configuration' screen to set a playback path if needed.
 2. This is accomplished by the ISD15100 SPI command: SPI_SND_DEC.

The added voice prompts will appear on the 'Voice Macro' screen for voice macro script editing.

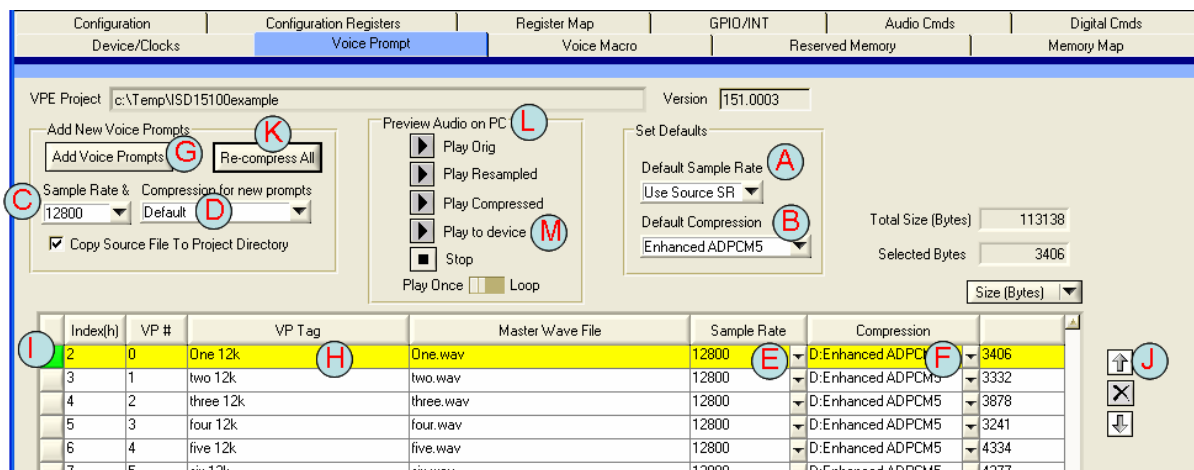


Figure 2-6 Voice Prompt screen

2.3.1 Keystrokes to Re-order the VPs

Besides using the up/down arrows (J) to re-order the VPs, users can also use the keystrokes:

- Select the VP by clicking the square (I).
- Press M on the keyboard and a pop-up box shows up as the Figure 2-7.
- Enter the new VP # to re-order the VP.

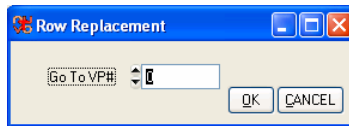


Figure 2-7 Re-order the VPs

2.4 Reserved Memory

The Recording Memory Pointer (RMP) divides the ISD15100 memory space into two blocks:

- Reserved Memory,
- Recording Memory.

The RMP is a two-byte address pointer pointing to a 4kbyte memory sector which is the first sector available for message recordings. Memory between address zero and the RMP pointer is considered the Reserved Memory for pre-recorded audio (Voice Prompts), pre-programmed macro scripts (Voice Macros), digital read/write access for other applications (User Data), and memory sectors reserved for the first sector of the reserved messages (Reserved Sectors). The memory between the RMP and the end of memory is considered the Recording Memory allocated for recording messages (Message Recordings).

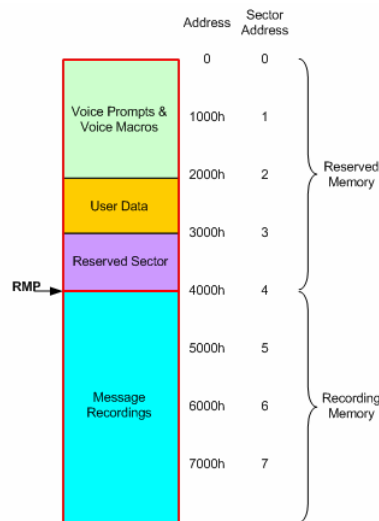


Figure 2-8 Example of ISD15100 Memory Map

On the 'Reserved Memory' screen, users can reserve memory sectors for message recordings:

- Empty messages (A):
 - o An Empty Message is an empty 4kByte memory sector reserved for message recordings. An SPI record command pointing to the Empty Message starts the message recording and will continue to record to the free memory sectors located in Recording Memory if the message recording is over 4kByte.
- Re-recordable messages (B):
 - o Users can have a default message when reserving the memory sectors for a message recording. For a re-recordable message, the first sector will be allocated in 'Reserved



Sectors' (refer to Figure 2-8); the rest of the message will be allocated in 'Recording Memory'.

Please note that only ADPCM, (D)uLaw, and PCM are supported for reserved messages, not Enhanced-ADPCM or VBR.

To record (or playback) a message, the microcontroller simply issues a record (or playback) message command with starting address pointing to the reserved initial address. The reserved messages are also listed on the 'Voice Macro' screen for voice macro script editing.

Memory sectors can also be assigned for digital read/write access for other application tasks (User Data). To reserve empty memory sectors, click the button 'Add Reserved Memory' (C) and then adjust the 'Number of Sectors' (D). To reserve (import) existing data, click the button 'Add Data' (E); the file size will be shown on 'number of sectors' (F).

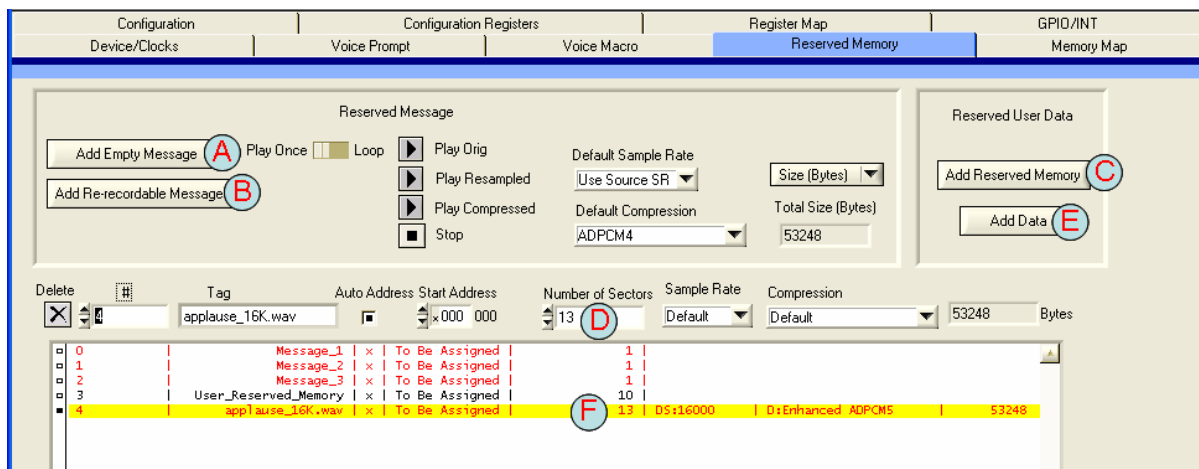


Figure 2-9 Reserved Memory

2.5 Voice Macro

Voice Macro is a powerful scripting feature integrated into the ISD15100. Voice Macro editor enables users to create scripted commands to play voice prompts (A), play reserved messages (B), insert silences (C), set the clock configuration (D), power down the ISD15100 (E), and configure the ISD15100 (F) such as setting a play or record signal path. As mentioned earlier, voice prompts (A) are from the 'Voice Prompt' screen, and the reserved messages (B) are from the 'Reserved Memory' screen, respectively. Each VM is assigned a tag and an index, where it can be executed with a single SPI command referring to the index.

There are two special Voice Macros:

- Index 0: VM0, Power-On Initialization (POI).
 - o Executed automatically when power initially supplied or when the ISD15100 chip is reset.
- Index 1: VM1, Power-Up (PU).
 - o Executed automatically when received power-up command.

POI and PU Macros allow users to customize how the part emerges from power-on or power-up conditions.

There are two special VM commands: FINISH (G) and WAIT INT (H).

- FINISH (G):
 - o Finish the voice macro and exit. Voice Macros must be ended with either FINISH or PD (power down).
- WAIT INT (H):
 - o Wait until current play command finishes before executing the next macro instruction.

The ISD15100 has a built-in command buffer so that consecutive commands can be executed one after another seamlessly. However, some commands don't go through the command buffer; therefore, they don't wait. Users may refer to the design guide; section 13 SPI Commands, Table 13-2, for details. In table 13-2 users may see that some commands can be executed regardless the status bit-1 CBUF_FULL (command buffer full).

WAIT_INT (H) is used whenever a command in a voice macro intends to wait to be executed until the previous play command finishes. A typical example will be adjusting the volume in a voice macro. Commands for adjusting the volume don't go through the command buffer; therefore, they will be executed immediately. Below is an example where the volume is immediately decreased when VP_A starts to play.

- Play VP_A
- Decrease Volume
- Play VP_B
- Finish.

If users want to decrease the volume after VP_A finishes, the voice macro should be modified as below:

- Play VP_A
- WAIT_INT
- Decrease Volume
- Play VP_B
- Finish.

Please note that the commands in a Voice Macro are executed sequentially, and all voice macros must be ended with PD or FINISH. The VPE15100 shows a warning message if it detects a voice macro not ended with PD or FINISH.

The example project has some VMs for demo purpose:

- VM0 (POI macro):
 - o Set the clock source to 4.096 MHz crystal.
 - o Set a playback path to PWM.
 - o Play a beep sound.
 - o Power down.
- VM1 (PU macro):
 - o Set the clock source to 4.096 MHz crystal,
 - o Set a playback path to PWM.
 - o Play two beep sounds.
 - o Idle.
- A few more voice macros for quality demonstration.

Below are the steps of how we make the POI macro:

1. Click 'Set CLK CFG' (D) to set the clock source to 4.096 MHz crystal (0x27).
2. Set CFG17 to 0x00 (F) so that registers won't be updated immediately.
3. Set CFG1A to 0x0F (F) so that GPIO pins 0 ~ 3 are set to output pins. Please note that the 7-segment display on the USB evaluation board is driven by the ISD15100 GPIO pins 0 ~ 3.
4. Switch to the 'Configuration' screen to set a playback path to PWM.
 - a. Click '007 PLAYBACK MEMORY-> PWM' (I). The path is shown in green on the path-view.
 - b. Click 'Sub-group selection' (J). Check the 'Audio Config' and 'UPDATE'.
5. Switch back to the 'Voice Macro' screen. Click 'Set All checked group' (K).
6. Click the voice prompt 'FastBeep' (A).
7. Click PD (E).

Other VMs are done in similar ways.

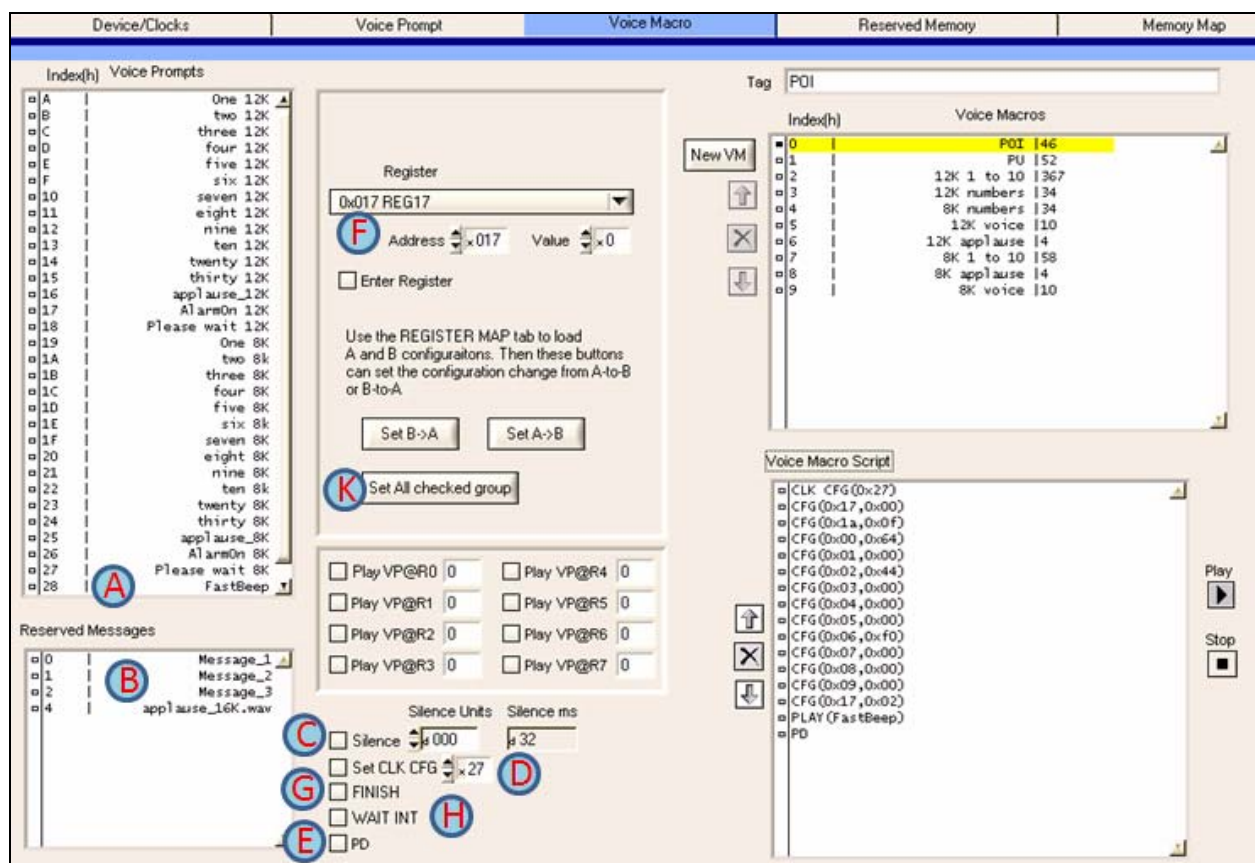


Figure 2-10 Voice Macro Editor

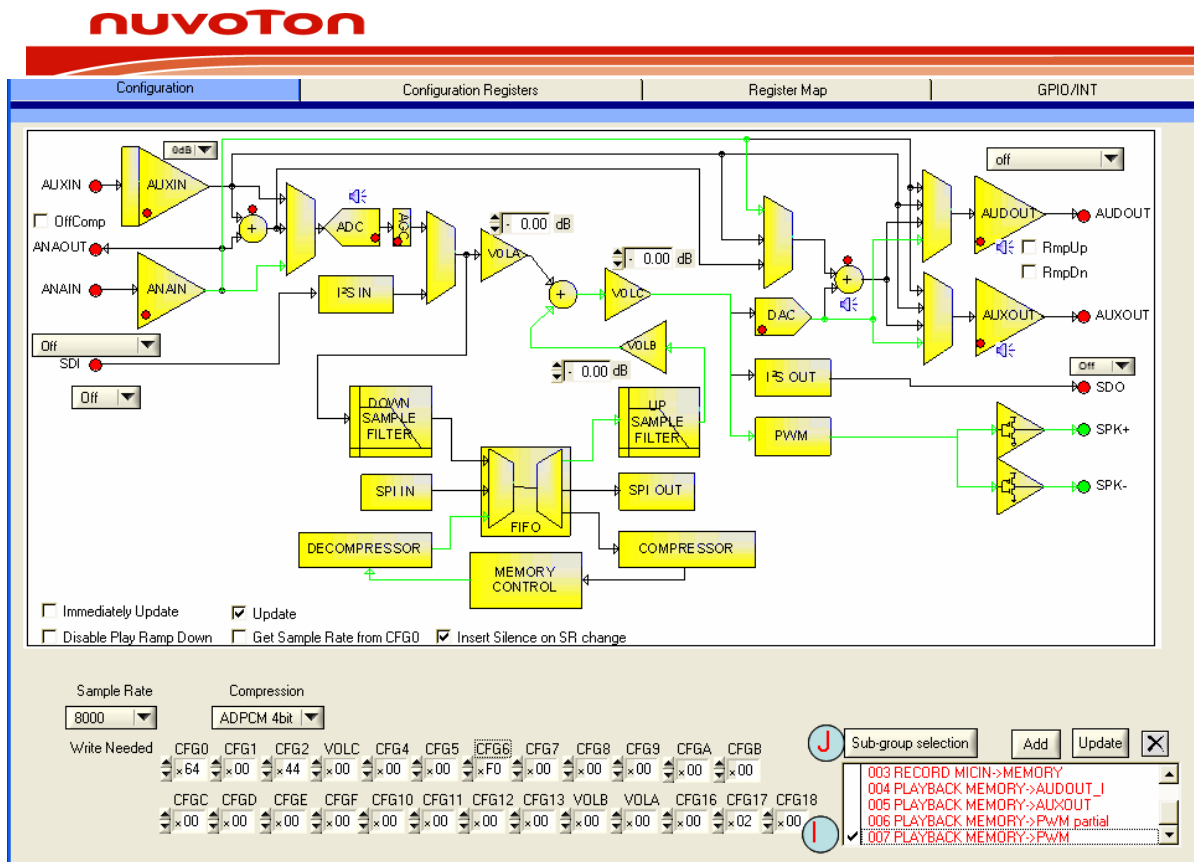


Figure 2-11 Set a Playback Path to PWM

2.6 Configuration

The Configuration screen aims to help users ease the task of configuring the ISD15100. To set the configuration registers, a microcontroller could either send several `WR_CFG_REG` commands or simply execute a Voice Macro that consists of a group of `WR_CFG_REG` commands.

Users can set a signal path for record or playback and generate automatic configuration bits that can be ported to the VM editor. Once a signal path is set, users can save it by clicking the 'Add' (A) and assigning a name for the path such as 'Play from Memory to Speaker', or 'REC from ANAIN'. The Configuration screen comes with several signal paths for users' reference. Please note that the signal paths that come with the Configuration screen are shown in red, while the user-defined configurations are shown in black. The Update (B) is used to update a user-defined configuration.

Please note that the CFG17 bit 0 (Immediately Update, C) and bit 1 (Update, D) are displayed on the path-view as these two bits decide how the configuration registers are updated.

- Immediately Update:
 - o This bit set to one turns off double buffering of configuration. Thus configuration is updated after each register byte is written. When this bit is set to zero, the configuration register setting is not made active until a one is written to the 'Update' bit.
- Update:

- o When a one is written to this bit and 'Immediately Update' is zero, configuration CFG0-CFG18 become active. In this way a path can be set up and applied at the same instant to prevent disturbance on the analog path (Write Only).

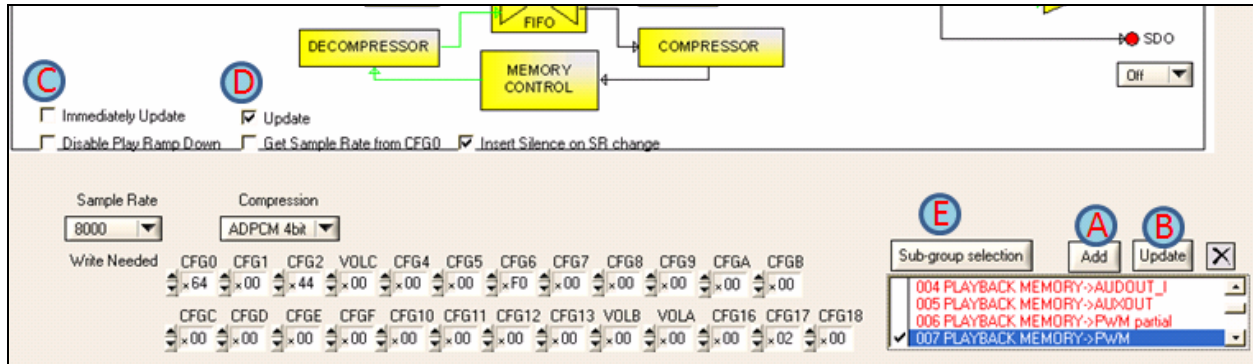


Figure 2-12 Configuration

'Sub-group selection' (E) groups the configuration registers into 7 categories:

Audio Config	REG 0x00 ~ 0x09
AGC	REG 0x0B ~ 0x0D, 0x0F
VOL	REG 0x14, 0x15
SPK_SEL	REG 0x18
UPDATE	REG 0x17
GPIO	REG 0x19 ~ 0x1D, 0x1F
VP at Rn	REG 0x20 ~ 0x2F

Table 2-3 Sub-Group of Registers

2.7 Memory Map

Creating the Memory Map is the last step to generate the binary programming file (the .mem file in the project folder). On the screen users will see how each VP, VM, and message is assigned an address and/or an index. Click 'Create Programming File' (A), and the memory map (B) is shown.

On the right hand side, the Memory Protection (C) decides how the memory contents are protected. The ISD15100 provides three kinds of protection mechanisms:

- Read Protection:
 - o Once enabled the digital-read command is not effective in the protected area.
- Write Protection:
 - o Once enabled the digital-write and memory-erase commands are not effective in the protected area.
- Chip-Erase Protection:
 - o Once enabled the chip-erase command is not effective.

The protection starts being effective after the ISD15100 chip is reset.

Please note that if both Chip Erase Protect (CEP) and Write Protect (WP) are enabled, the memory contents will be locked from any further modifications.



The check boxes under the 'Project Information' (D) allow users to add project information to the binary programming file. If users check the 'User Comment' box, whatever is typed in will be appended to the end of the project information.

If the USB evaluation board is connected to PC, the 'Burn Device' button (E) will be active. Once the binary programming file is generated, users can click the 'Burn Device' so that the VPE15100 will:

1. Chip-erase the ISD15100,
2. Program (digital-write) the binary programming file into the ISD15100,
3. Reset the ISD15100.

Please note that 'Date' (F) is part of the project, so clicking the 'Create Programming File' (A) may change the project contents even though users don't change anything. When users open an existing project, the memory map will be loaded automatically.

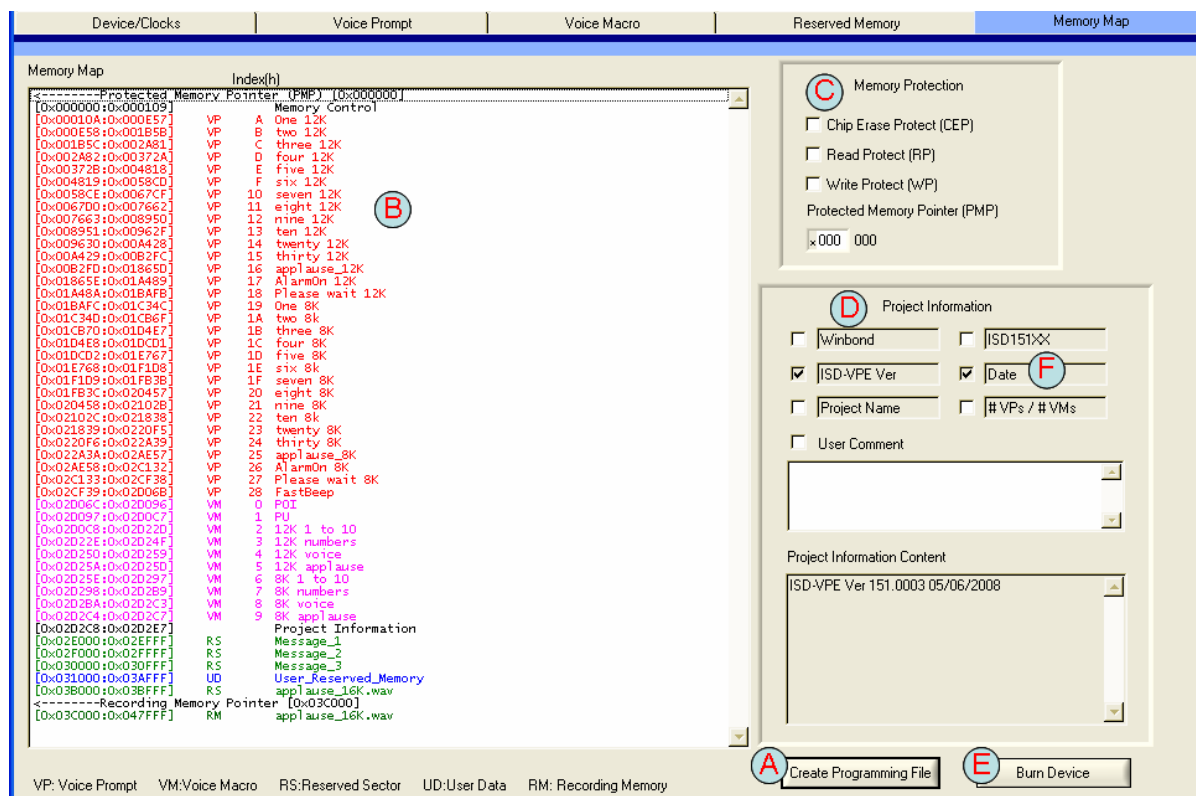


Figure 2-13 Memory Map

2.8 Archive and Clean

Under the Project menu there are two useful functions:

1. Archive
2. Clean

These two functions are used to backup the project files and clean unnecessary files. They are useful when users want to send an VPE15100 project to other people.

The Archive function duplicates a project to an archive directory with exactly same project name and structure, and changes the project links to point to this directory so that all paths are stored relative to the project directory. For example, if users create a VPE15100 project in a network share folder which is, say, mapped to T drive, users cannot just send this project to other people because "T" drive means nothing to other people. By archiving the project, that project is now self contained and can just be copied to another machine or to backup.

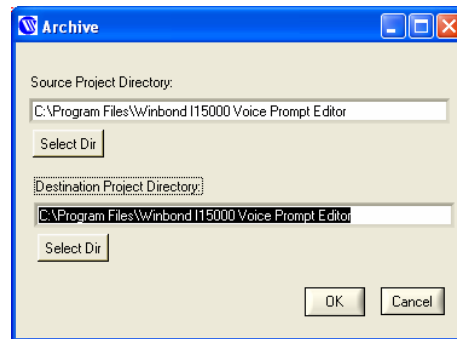


Figure 2-14 Archive

The Clean function deletes the contents in the BinFile and WavFile folders as they can be regenerated. The WavFile folder contains wave files of re-sampled and de-compressed voice prompts. The BinFile folder contains files of compressed voice prompts.

When a project is created, three sub-folders are created:

- BinFiles: where the compressed VPs are stored.
- WavFiles: where the re-sampled and de-compressed VPs are stored.
- WavSrc: where wave source of VPs are stored.

A project file has the .prj extension. The binary programming file for the project has the .mem extension. A header file with the .txt extension is also included for C language firmware programming.

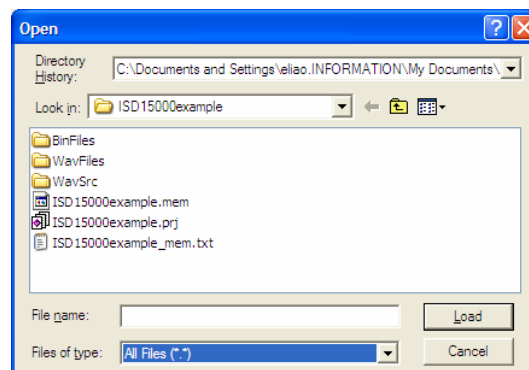


Figure 2-15 VPE15100 Project

3. Evaluation Board Overview

Figure 3-1 below shows the ISD15100 USB evaluation board.

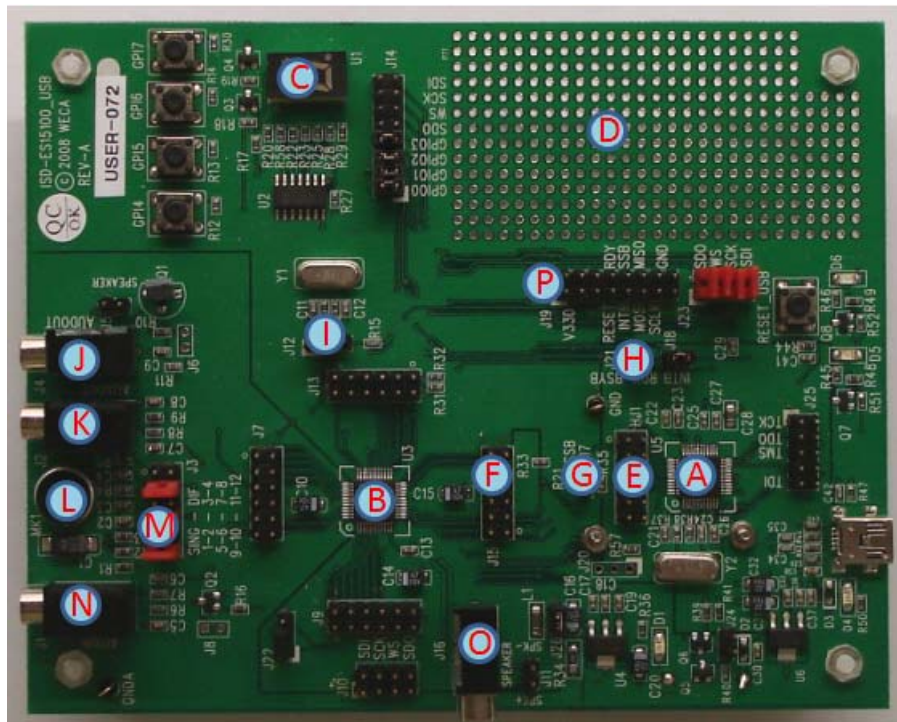


Figure 3-1 ISD-ES15100_USB

On the right hand side is the ISD15U110 (A), a USB1.1 controller especially designed for ChipCorder ISD15100 family. In the middle is the ISD15108 (B), an 8-minute part based on 8 kHz & 4-bit ADPCM. On the left hand side are analog circuitry and connectors for recording and playback. Top left is a 7-segment display (C) controlled by the ISD15108 GPIO pins 0 ~ 3. Top right is a hole-array (D) for users' prototyping.

The USB evaluation board can be operated standalone or controlled by the VPE15100. To operate standalone, make sure that the VPE15100 project has been properly programmed into the board; a playback path must be set in users' PU macro (VM1). Attach the keypad onto the pin-header HJ1 (E) and insert 3 AAA batteries into the battery holder underneath the board. The USB controller will power-on and reset the ISD15108 in 3 seconds.

Press the buttons to:

- ExeVM: power-up the on-board ISD15108 chip or send the ExeVM command.
 - o The USB controller automatically checks if the ISD15108 chip is powered up. If not, it sends the power-up command. If yes, it sends the ExeVM command.
- STOP: send the STOP command and clear index.
- 0 ~ 9: indexes for executing VM.
- VOLUP/VOLDN: increase/decrease volume.

For example, to execute VM 25, do the following:

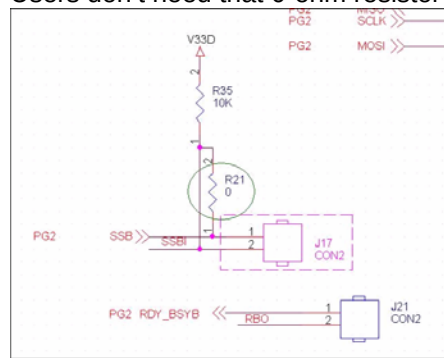
- Press ExeVM. The on-board ISD15108 chip will be powered-up.
- Press 2. Users shall see '2' displayed on the 7-segment display (C).
- Press 5. Users shall see '5' displayed on the 7-segment display (C).
- Press ExeVM to execute VM 25.



Figure 3-2 Keypad for Standalone

Around the on-board ISD15108 (B) there are four pin-headers (F) used for adaptation of daughter cards. To put any daughter card onto the USB evaluation board:

1. Make sure the on-board ISD15108 chip is chip-erased.
2. Remove the two jumpers J17 SSB (G) and J21 RDY_BSYB (H).
 - o In case the 0-ohm resistor R21 (next to the jumper J17 SSB) is on the board, remove it. Users don't need that 0-ohm resistor as it is fully controlled by the jumper J17.



3. Put the daughter card onto the four pin-headers (F).

On the board there are a 4.096 MHz crystal and an 80k-ohm resistor for clock setting. Users use jumper J12 (I) to select the clock sources:

- Internal oscillator with internal resistor:
 - o It doesn't matter how the jumper J12 (I) is set.
- Internal oscillator with external resistor:
 - o Put jumper J12 (I) on 2-3.
 - o Users can change the external resistor for different sampling rate.
- Crystal:
 - o Put jumper J12 (I) on 1-2.
 - o Users can change the crystal for different sampling rate.

On the left hand side are analog circuitry and connectors for recording and playback:

- Aud-out (J): can be configured as either a current-type or voltage-type single-ended output.
 - o When configured as a voltage-type output, it is exactly the same as the aux-out.
- Aux-out (K): a voltage-type single-ended output.

- o Users need an external amplifier to drive the speaker.
- Mic-in/Ana-in (**L**): can be configured as either a differential or single-ended input.
 - o Install jumpers (**M**) on 3-4, 7-8, and 11-12 for differential input (Mic-in).
 - o Install jumpers (**M**) on 1-2, 5-6, and 9-10 for single-ended input (Ana-in).
 - o Please note that the USB controller uses a general I/O pin to “physically” enable/disable the microphone. Later when users connect their own microcontroller to control the ISD15100, they can either:
 - Manually enable the microphone by installing the jumper on J8 (this increases the standby current), or
 - Use an I/O pin to “physically” enable/disable the microphone as the USB controller does.
- Aux-in (**N**): a voltage-type single-ended input.

PWM (**O**) is a direct speaker drive.

It can also be configured as a voltage-type differential output (BTL):

- BTL can directly drive the speaker, but the volume level is not as loud as PWM. Users can use an external amplifier to increase the volume of BTL.

The two pins can also be separately configured as Aud-out/Aux-out.

Below is a table for the ISD1510 evaluation board configuration.

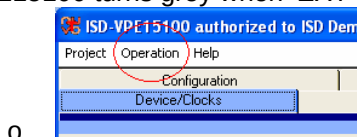
Jumper Name	Function	Default
J17, J21	• Take off J17 & J21 when daughter card is used. - o R21 must be removed. • Install J17 & J21 when daughter card is NOT used.	J17 & J21 are installed to de-select the daughter card.
J12	• Install jumper on 1-2 to select the crystal as the clock source. • Install jumper on 2-3 to select the external resistor as the clock source.	Jumper is installed on 1-2 to select crystal as the clock source.
J3	• Install jumpers on 1-2, 5-6, and 9-10 to configure as Ana-in (single-ended). • Install jumpers on 3-4, 7-8, and 11-12 to configure as Mic-in (differential).	Configured as Mic-in (differential input).
J10	• I2S Connector (SDI,SCK,WS, SDO)	

Table 3-1 EV Board Configuration

3.1 Connect a uC to the Evaluation Board

Users can use their own microcontroller to talk to the on-board ISD15108 chip by following the steps below:

- Connect the USB evaluation board to PC and then launch the VPE15100.
- On menu bar: Operation -> select 'EXT Control' so that the USB controller releases the SPI bus. VPE15100 turns grey when 'EXT Control' is selected.



o

- The microcontroller can then talk to the on-board ISD15108 via the pin-header J19 (P).
 - Users need at least the four SPI signals plus a common ground. If users intend to implement digital-read or digital-write commands, they need to poll RDY/BSYB pin for data flow control as well. All controlling signals are available at J19 (P).
- Always start with Read_Status:
 - We encourage the users to start with the Read_Status command because this command is always needed for debugging.

3.2 Program the ISD15100

There are several options to program the ISD15100:

1. Hi-Lo Systems Taipei <http://www.hilosystems.com.tw> can provide the programming service for the ISD15100. Below are some more details:
 - Their programming center is in Taiwan.
 - The programming charge is USD \$0.05 ~ \$0.07 per sample depending on quantity.
 - We are not responsible for it if the price offered by Hi-Lo is different.
 - We will not get involved. It is between the customers who are buying the ISD15100 and the programming companies.
 - Below is the contact information:
 - Lisa Huang
 - TEL: 886-2-8792-3301
 - Email: lisa@ms1.hilosystems.com.tw
2. Hi-Lo Systems <http://www.hilosystems.com.tw> has made a 4-socket gang programmer for the ISD15100.
 - All-100
 - M4-ISD15000-QF48
3. Infonitek <http://www.infonitek.com.tw/plr.htm> has made a 6-socket gang programmer for ISD15100.
 - IT2000
 - IT2015T
4. We have a single-socket programmer adaptor board that has to be used together with the ISD15100 evaluation board.
 - ISD-ES15100_PROG (Figure 3-3)

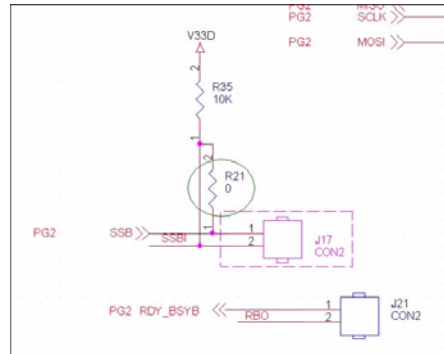


Figure 3-3 ISD-ES15100_PROG

3.3 The Programmer Daughter Card – ISD-ES15100_PROG

To use the programmer daughter card, ISD-ES15100_PROG, please follow the steps below:

1. Chip-erase the on-board ISD15108 chip.
 - o This is to prevent the POI macro of the on-board ISD15108 chip from conflicting with the ISD151xx chip in the daughter card.
2. Remove the two jumpers J17 SSB (G) and J21 RDY_BSYB (H).
 - o In case the 0-ohm resistor R21 (next to the jumper J17 SSB) is on the board, remove it. Users don't need that 0-ohm resistor as it is fully controlled by the jumper J17.



3. Put the daughter card onto the four pin-headers (F).

3.4 The Demo Board – ISD-DEMO15100

ISD-DEMO15100 is a supplemental board to the USB evaluation system ISD-ES15100_USB. The goal of the ISD-DEMO15100 is to help customers build a standalone system of the ISD15100 with their own microcontroller for demonstration purpose.



Figure 3-4 ISD-DEMO15116

This small demo board could be connected to the ISD-ES15100_USB for controlling and programming.

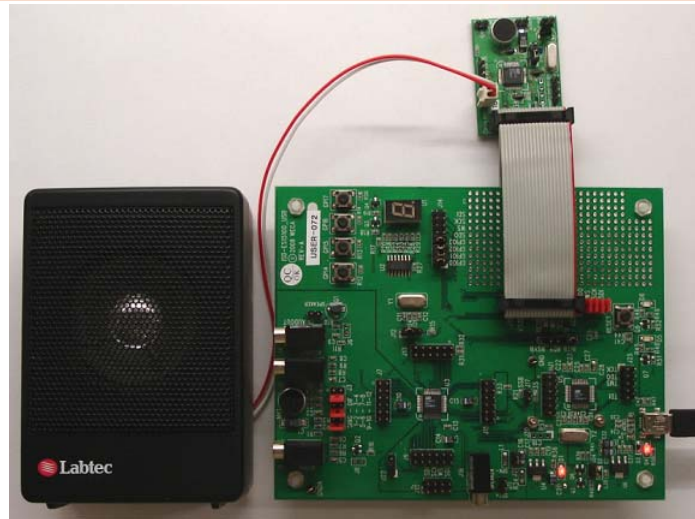


Figure 3-5 ISD-DEMO15100 connects to ISD-ES15100_USB

For details please refer to the user's manual of the ISD-DEMO15100.

4. USB Emulation

When the USB evaluation board is connected to PC, two extra screens show up:

- Audio Cmds:
 - o Where users can emulate the ISD15100 play and record commands like executing voice macros, erasing recorded messages etc.
- Digital Cmds:
 - o Where users can emulate the ISD15100 digital commands like downloading a project, reading back for verification etc.

4.1 Audio Cmds

The ISD15100 provides two message recording mechanisms:

- 'Record Msg' (A):
 - o Record a message without specifying the address. The ISD15100 automatically searches for available memory sectors starting from the RMP (Recording Memory Pointer) for the message recordings. When a 'Stop' command is issued to finish the recording, or the recording stops due to memory full, the VPE15100 automatically reads back the start address and length of the recorded message.
- 'Record Msg At' (B):
 - o Record a message at the user-specified address, which is reserved in advance. By default, the reserved messages are allocated before the RMP. Since the start address of the Reserved Message is allocated in advance, there is no need to read back the start address of the recorded message. The Reserved Message is especially useful for a predefined scenario.

The 'Play Msg At' command (C) plays a message at the user-specified address. Users can play a message from the middle by specifying an offset (D). One offset is one memory sector, which is about one second based on 8kHz 4-bit ADPCM. The ISD15100 does not provide a PAUSE command; however, a PAUSE command can be easily done by toggling between 'Stop' and 'Play Msg At' with an offset.

To erase a message, simply specify the start address and then click 'Erase Msg At' (E). Addresses of recorded messages are shown on the Message List. (F)

'Get Msg List' (G) is not an ISD15100 command. It is indeed a string of digital-read commands used to scan the memory for messages. The first byte of each memory sector shows whether or not a 'beginning of message (BOM)' is inside the memory sector.

VM (H) & VP (I) sub-screens show users the VM/VP tags and indexes that they are working on.

To execute a VM or to play a VP, simply click the VM/VP sub-screens to get the index, and then click 'Execute VM' (J) or 'Play VP' (K).

VM and VP are index based, which means to execute a VM or to play a VP the microcontroller only sends a simple command followed by the index. They are address-free! Please note that VM and VP share the indexes, and the index always starts from VM. For example, if a project has three VMs and four VPs, then the indexes 0 ~ 2 belong to VMs, and the indexes 3 ~ 6 belong to VPs.

The 'Indirect Voice Prompt' (L) is to make the voice macro flexible. For example, it could be used to announce time and temperature in users' system. To announce a sentence like '*It is 7:15 am; the outside temperature is 70 degree Fahrenheit*', users could do the following:

- The microcontroller gets the time (7:15 am) and temperature (70 degree Fahrenheit) from the system's timer and sensor.
- The microcontroller writes the corresponding voice prompt indexes of 7, 15, am, and 70 into R0, R1, R2, and R3.
- The microcontroller then executes the voice macro of that sentence.

To emulate it on the VPE15100, click the VP sub-screen (I) to get the index, then right-click Rn, n = 0 ~ 7, to update Rn, then click 'Play VP@Rn' (L).

The volume control bar (M) provides an easy way to adjust the volume while playing a VP or a message. ISD15100 provides three volume controls:

- VOLA: volume control for the digital audio data from the I²S or analog inputs.
- VOLB: volume control for the digital audio data from the decompression block or SPI.
- VOLC: master volume control for PWM, Aud-out, Aux-out, and I²S outputs.

Resolution of the three volume controls is 0.25dB per step, total 256 steps for each volume control. The volume control bar here is the VOLC. What the VPE15100 does is keep writing the CFG3 when the users drag the volume control bar.

'Mic-in/Ana-in Enabled' (N) indicates that the microphone path is enabled. When the microphone path is enabled (either configured as differential or single-ended), the indicator turns red.

The ISD15100 provides four 'SPI Data Transfer' commands. For details, please refer to design guide. Take 'SPI Decode' (O) as an example, the microcontroller sends the compressed audio data (VPs or recorded messages) to the ISD15100 to decode. Double-click 'PLAYBACK MEMORY->PWM' (P) to set a playback path for 'SPI Decode'. Check 'CTS SPI' (Q) if users want a continuous SPI transaction, which means the VPE15100 ignores all other tasks before the 'SPI Data Transfer' command finishes. Right click the Filename blank (R) to assign a compressed audio file, and then click 'SPI Decode' (O).

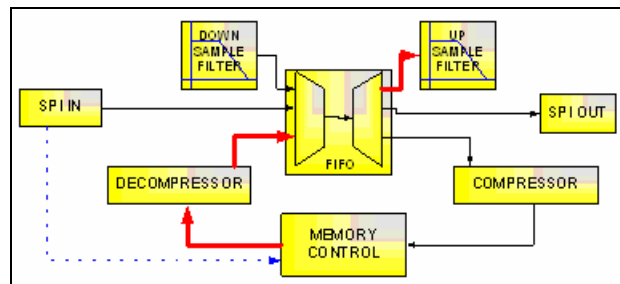


Figure 4-1 SPI Send Compressed Data to Decode

Users can try different clock settings (S). Please note that the clock setting must match the real configuration on the board or the ISD15100 won't work. Below is a table of some common settings:

0x34	Internal oscillator with internal resistor
0x36	Internal oscillator with 80kohm external resistor
0x37	2.048 MHz crystal
0x27	4.049 MHz crystal
0x17	11.2896 MHz crystal

Table 4-1 Some Common Clock Settings

'Read ID' (T) reads back the four-byte ISD15100 ID.

'Read Status' and 'Read INT' (U) both read back the two-byte ISD15100 status. The only difference is that 'Read INT' clears the pending interrupt as well. The ISD15100 will not generate a new interrupt signal if there is a pending one.

'Saved Configurations' (P) is a duplicate from the 'Configuration' screen. Double-click a setting to write to the ISD15100.

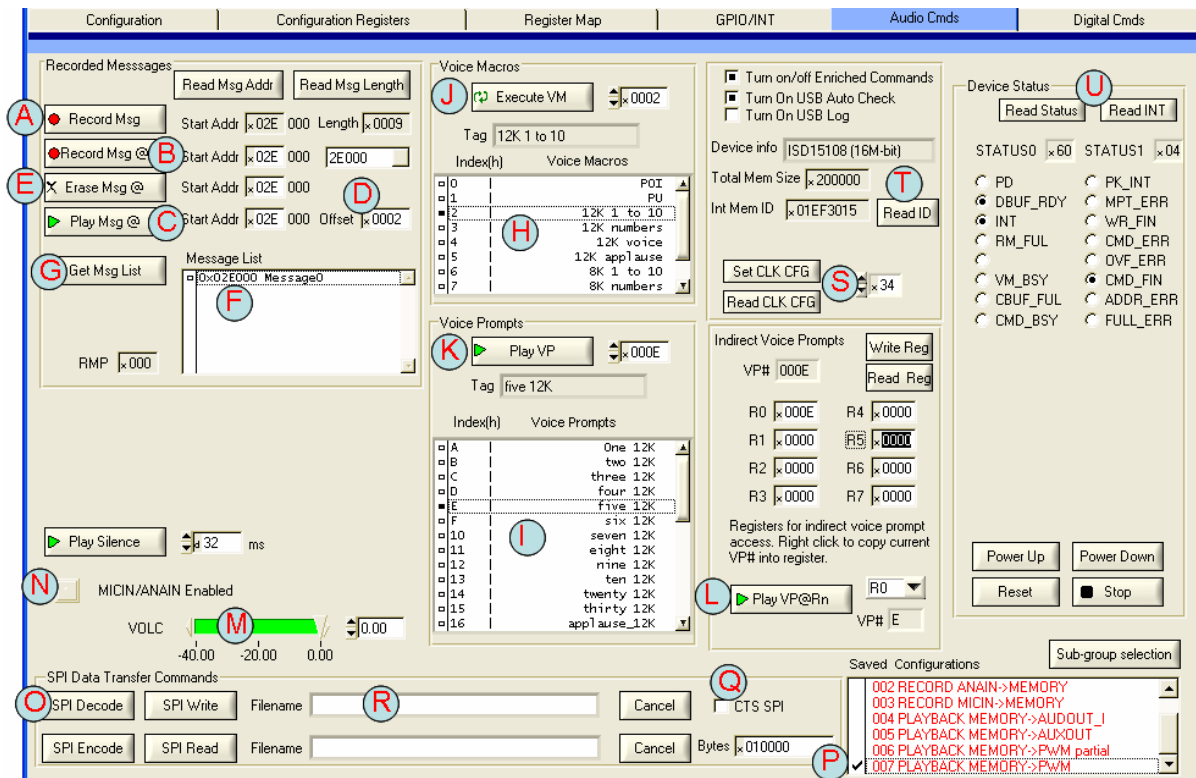


Figure 4-2 Audio Cmds

4.2 Digital Cmds

The ISD15100 provides four digital commands: digital-read (A), digital-write (B), erase-memory (C), and chip-erase (D).

To erase the whole memory, click 'Chip Erase' (**D**); while the erasing is in progress, status bit CMD_BSY (**E**) goes high. Time for erasing the chip depends on the memory density. Below is a table for reference.

Device	Memory Density	Typical Erasing Time
ISD15102	4Mbit	5 sec
ISD15104	8Mbit	10 sec
ISD15108	16Mbit	25 sec
ISD15116	32Mbit	40 sec
ISD15132	64Mbit	80 sec

Table 4-2 Chip Erasing Time

To erase partial memory, specify the start (**F**) and end addresses (or start address and the number of sectors to erase), and then click 'Erase Memory' (**C**); while the erasing is in progress, status bit CMD_BSY (**E**) goes high. The minimum erasable memory size is one sector (4k bytes).

Memory contents can be read back and either displayed on screen or saved as a file on PC.

To display the memory contents on screen:

- Specify the start (**G**) and end addresses (or start address and the number of bytes to read).
- Specify to display on screen (**H**) (the default).
- Click 'Digital Read' (**A**).

To save the memory contents on PC:

- Specify the start (**G**) and end addresses (or start address and the number of bytes to read).
 - o Users can click 'Read ID' (**I**) to get the memory size.
- Specify to save as a file (**H**). Click 'Select File to Save' (**J**) to specify the path and filename.
- Click 'Digital Read' (**A**).

Users can update the memory contents of the on-board ISD15108 by digital-writing a file on PC or by writing the data bytes shown the screen.

To update the memory contents from a file:

1. To erase the whole chip and then download a file from PC:
 - o Check 'Mass Erase/Prog' (**K**).
 - o Specify to update from a file (**L**) (the default). Click 'Select File to Open' (**M**) to specify the path and filename.
 - o The start (**N**) and end addresses will be loaded automatically.
 - o Click 'Digital Write' (**B**), and the VPE15100 will chip-erase the whole memory and then download the file to the ISD15100.
2. To overwrite partial memory contents by a file from PC (the default):
 - o Uncheck 'Mass Erase/Prog' (**K**).
 - o Similar procedure as the above.
 1. The VPE15100 accomplishes this task by using PC's resources. For users' reference, below is how the USB controller does the task.
 - If needed,
 - o Read back one 4kbyte sector and save on PC's buffer.
 - o Overwrite the buffer with the corresponding portion of the file.
 - Erase one memory sector of the on-board ISD15108.
 - Program the buffer on PC to the ISD15108 memory sector that is just sector-erased.
 - Repeat the above steps.

To update the memory contents from the screen:

- Read out the ISD15100 data and have it displayed on screen as described above.
- Edit the data contents on screen.
 - The first sector (the sector 0) is the ISD15100 memory header. Do not mess it up.
- Specify to update the memory contents from screen (L).
- The start (N) and end addresses will be automatically matched.
- Click 'Digital Write' (B).
 - The VPE15100 accomplishes this task by using PC's resources.

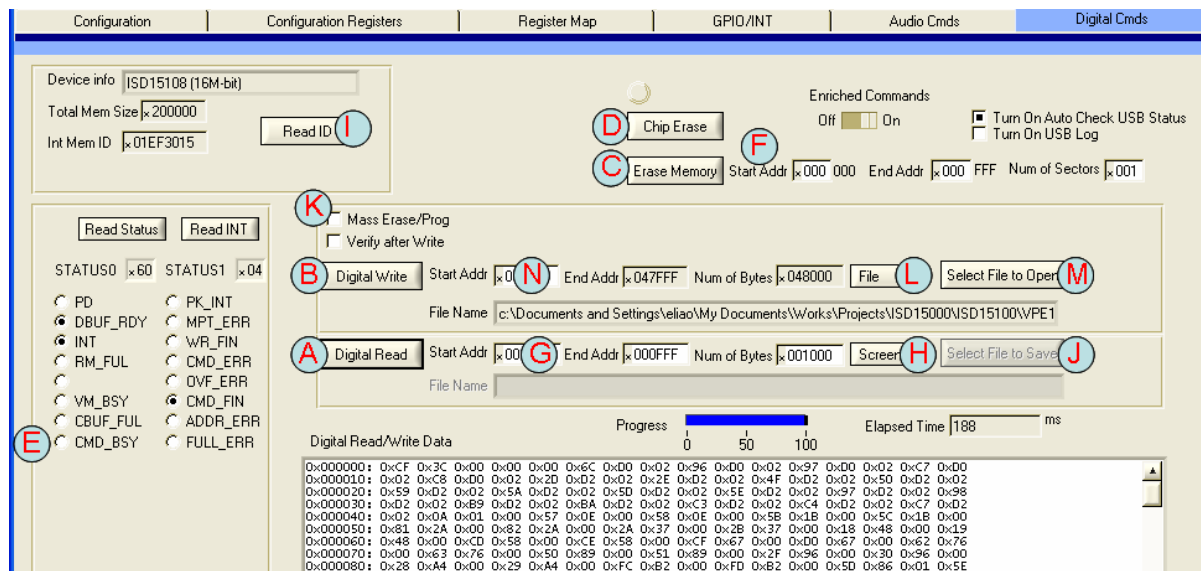


Figure 4-3 Digital Cmds

4.3 Enriched Commands & USB Logs

Enriched-command mode is to make the VPE15100 user-friendly. It adds extra commands for the users:

- Check the ISD15100 status before issuing a command.
- Check signal path setting before issuing a record or playback command

If users do not want the VPE15100 to add extra commands, simply turn it off.

USB Logs show users how the USB controller controls the ISD15100. Click 'Turn On USB Log' and the transactions between the ISD15100 and ISD15U110 are logged in a file located in the VPE15100 installation folder, which by default is "c:\Program Files\ISD-VPE15100\Logs".

Below is a log example of executing VM 2 with enriched on and auto-status-checking disabled:

/-----

Execute VM:

ReadStatus: 0x40 0x00 Status Readback: 0x60 0x04

ReadCfgReg: 0xBA 0x02 0x00 CFG Readback: 0x44

ReadINT: 0x46 0x00 Status Readback: 0x60 0x04

ExeVM: 0xB0 0x00 0x02

Below is a log example of executing VM 2 with enriched off and auto-status-checking disabled:

```

/-----
Execute VM:
ExeVM:      0xB0 0x00 0x02
\-----

```

Since the enriched is off, the VPE15100 does not check the ISD15100 status or signal path setting before issuing the command.

5. Revision History

Version	Date	Description
151.0003	May 21, 2008	Initial release:
151.0014	Mar 16, 2009	Update: • Contents of the kit. Add: • Usage of the ISD-ES15100_PROG and ISD-DEMO15100. • Sub-group of the registers. • Keystrokes for reordering VPs.